

Project Canasta

Jeffrey Wang

SMWCon – 9 December 2021



Important disclaimer

Please note these slides are accurate as of December 9, 2021, when this presentation was given.

Certain information has been edited from the version originally presented. PII has been redacted.

This presentation is © 2021 WikiTeq, Inc. and MyWikis LLC. All rights reserved.

History

Early history

- Started in late 2020
- Four goals
 - Easy to set up and configure MediaWiki
 - Easy to run MediaWiki with Apache, MySQL, PHP, Elasticsearch, etc.
 - Ability to install wiki pages on the resulting wiki
 - Wiki starter page packages for various use cases
- Proposals such as Docker, Ansible, etc. were floated

Enterprise MediaWiki Hackathon 2021

- Many Canasta ideas formalized
 - Use Docker
 - Options for Docker Compose and Kubernetes
 - Extensions to include
 - General structure of the image



Dallas, Texas, USA
May 24-26, 2021

Work since the hackathon

- Developed first iteration of Canasta Docker image and stack
- Current development work contributed by WikiWorks, WikiTeq, and MyWikis
- In use by some WikiWorks and WikiTeq clients
- Working on general availability release in the near future

Background

MediaWiki for all

- Easy to set up, hard to set up well
- Well-oiled MediaWiki instance requires lengthy setup and constant maintenance
- Difficult to achieve enterprise and organizational use cases of MediaWiki without extensive efforts

Common pitfalls of installing MediaWiki

- No short URL by default
- Need to research and download many extensions
- Non-functioning job queue
- No CirrusSearch
- Error-prone upgrade process

Design

Goals

1. Beginner friendly
 - a. Easy and fast to install, upgrade, and maintain
2. Convenient
 - a. Everything's already present, all-in-one package
3. Same core MediaWiki
4. Stable
5. Customizable
6. Extensible
7. Source control compatible

Canasta stack

- Web application, comprised of
 - MediaWiki
 - Apache and PHP
 - Job queue runner
 - Maintenance script runner
 - Automated database backups
 - 100+ bundled extensions
- MySQL
- Elasticsearch
- Choice of Docker Compose or Kubernetes

Bundled extensions and skins

- 100+ extensions bundled
 - Visual Editor, Page Forms, Semantic MediaWiki, Cargo, etc.
- Several skins bundled
 - Vector, MonoBook, Timeless, Chameleon
- Updates handled by Canasta
- Additional extensions and skins, including custom ones, can still be used
- Bundled extensions and skins can be included with `cfLoadExtension()` and `cfLoadSkin()`
 - Same syntax as `wfLoadExtension()` and `wfLoadSkin()`
 - Allows choosing between including bundled or custom extension

Flexible LocalSettings.php

- Existing LocalSettings.php can be drag-and-dropped
- Built-in configuration for MySQL in Canasta stack coming soon
 - Recommended to remove \$wgDB variables from LocalSettings.php
- Two options for loading settings
 - Traditional LocalSettings.php file
 - LocalSettings.php.d/ directory

Impact

Advantages

- Quick to set up
- Low maintenance
- Feature rich
- No configuration drift
- Easy to be compatible with future versions of MediaWiki
- Scalable and performant

Scenarios

Comparing scenarios in vanilla MediaWiki to Canasta

- Setting up MediaWiki
- Upgrading MediaWiki
- Adding/enabling an extension or skin
- Backing up database
- Load balancing
- Deploy on the cloud: ECS, Fargate, Azure Container Instances, EKS, AKS, etc.

Demo

Installing and running Canasta

1. Have Docker Engine running on your server, being sure port 80 and 443 are not in use
2. Git clone Canasta stack repo
3. Copy .env.example to make your own .env
4. Run docker-compose up -d
5. Follow MediaWiki setup, use these database settings:
 - a. Host: db
 - b. Database name: mediawiki
 - c. Username: root
 - d. Password: (whatever you set it to)
6. Place LocalSettings.php in config/ folder
7. Done!

Caveat:

Currently there is a small bug with skins. To fix this, go into the skins/ directory and make the following symlink:

In -s /var/www/mediawiki/w/canasta-skins/Vector

(You could replace Vector with MonoBook, etc.)

Visit a working Canasta test instance:
[Redacted]

What's next

Visit a working Canasta test instance:
[Redacted]

Release info

- Open source, available on GitHub at github.com/WikiWorks/Canasta
- Project Canasta to be ran by its own governance structure
 - In the meantime, led by Yaron Koren
- General availability coming soon (no release date yet)

Visit a working Canasta test instance:
[Redacted]

Future development goals

- Finalize specifications for Canasta 1.0
- Automatic HTTPS via Traefik or Caddy container
- “Manager” image separate from Canasta web app image
- Create starter pages in Page Exchange format
- Add options for even beefier setups
 - Memcached, Redis
 - Varnish
 - etc.

Visit a working Canasta test instance:
[Redacted]

More info

Jeffrey Wang (MyWikis)

[Redacted]

Also on MWStake Matrix

Yaron Koren (WikiWorks)

Ike Hecht (WikiTeq)

Visit a working Canasta test instance:

[Redacted]



Zion National Park
Utah, USA
November 11, 2021